UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/605,465	10/01/2003	Ivan Iliev Klianev		2464

38345   7590   05/17/2007

EXTRAPOLES PTY LIMITED
PO BOX 324
BRIGHTON-LE-SANDS   NSW, 2216
AUSTRALIA

EXAMINER
YAARY, MICHAEL D

ART UNIT	PAPER NUMBER
2193	

MAIL DATE	DELIVERY MODE
05/17/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *23 February 2007*.

2a)☒ This action is **FINAL.**   2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-3 and 5-20* is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-3 and 5-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All   b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
    Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

Detailed Action

1.   Claims 1-3 and 5-20 are pending in the application.

Claim Rejections - 35 USC § 101

2.   35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

3.   As to claims 5-17 the claims are rejected under 35 U.S.C 101 because the claimed invention is directed to non-statutory subject matter.

(i)   As to claim 5, the system is software per se, as it is not tangibly embodied, failing to recite any hardware as part of the system. See suggested examiner amendment attached below for suggested correction.

(ii)   Claims 6-17 are rejected for similar reasons as discussed for their respective parent claim, as they fail to present any limitations that resolve the deficiencies of the claim from which they depend.

Claim Rejections - 35 USC § 112

4.   The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

5.   Claims 1, 5, and 18-20 are rejected under 35 U.S.C. 112, second paragraph, as the claims are vague and indefinite as to the claimed subject matter.

(i)   As to claim 1, it is unclear in step (E) (F) and (G), whether the compiling and linking instructions are compile and linked a single time or multiple times in the way the steps are written. If so, are steps (F) and (G) necessary, or are the redundant steps. Claim 1 is also directed to a "method for graphical development (line 1)," yet claim 1 discloses nothing teaching graphical implementation making the intention of the claim unclear.

(ii)   As to claim 5, part (B), "where necessary," provides an indefinite and uncertainty as to where the workflow-synchronization object would be necessary. Part (E) of claim 5 also provides uncertainty using the optional, as it is unclear when the third level's sibling would be required.

(iii)   As to claims 18-20, the claims are written improperly as being directed to a computer storage medium, when they claim dependency on multiple claims whose parent claim are method and a system. See suggested examiner amendment below to resolve the issues, as examiner suggests claim 18 to be an independent claim.

Claim Rejections - 35 USC § 102

6.   The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that

form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

6.   Claims 5, 19, and 20 are rejected under 35 U.S.C. 102(b) as being anticipated by

Du et al (hereafter Du)(US Pat. 6,308,163).

7.   Du was cited in the previous office action dated 12/14/06.

8.   **As to claim 5,** Du discloses a system for transactional processing of workflow,

wherein skeleton of said system is a hierarchical tree of class objects with capacity to

represent variety of workflow configurations (column 6, lines 62-65), wherein said class

objects are computer memory instances of classes or structures (This is inherent as

class objects correspond directly to a block of computer memory.), wherein any level of

said hierarchical tree contains one or multiple collections (column 8, lines 3-7), each

said collection containing predefined or dynamically defined number of one or multiple

objects (column 8, lines 5-7), wherein objects belonging to any said collection of same

hierarchy level are computer memory instances of classes or structures of identical type

(This is inherent as class objects correspond directly to a block of computer memory.),

wherein first level of said hierarchical tree consists of a single collection (column 7, lines

8-11) and every next level of hierarchy contains one or multiple collections and possibly

individual objects (system 36 of figure 2 and column 7, lines 12-13), wherein said

collections and said individual objects being accessible via references controlled by

objects belonging to collections from previous level (column 7, line 21 and line 42

disclose levels interacting with previous levels), wherein said hierarchical tree of class

objects contains following levels and siblings:

(A) First level, a steps-collection, wherein said collection objects represent steps

of said workflow-process, each said object being parent of a Second level collection

(column 7, lines 8-11 disclose a first level providing an overview of the processing

capabilities of the workflow resource system);

(B) Second level, an activities-collection per step of said workflow-process,

wherein said collection objects represent workflow-activities that might be executed

concurrently with all other workflow-activities represented by objects belonging to same

collection (column 7, lines 12-14), wherein each said object is parent of Third level

siblings, said siblings being of 3 distinct categories: a processor collection, a workload-

balancer object, and where necessary a workflow-synchronization object;

(C) Third level's sibling One, a processors-collection per workflow-activity,

wherein an object of said collection represents an individual workload-processing squad

having a capacity to process a portion of entire workload specific for said workflow-

activity, each said object being parent of a Fourth level collection;

(D) Third level's sibling Two, a workload-dispatching object per workflow-activity;

(E) Third level's optional sibling Three, a workflow-synchronizing object per

workflow-activity if required by flow-graph of said workflow-process, wherein said object

contains data defining required synchronization scheme (Column 7, lines 25-41 disclose

a third level of multiple local resource managers (LRMs, which would be sibilings)

controlling and managing the different resources, or workflow processes. This therefore,

just an obvious variation of the third-level siblings of the instant application.);

(F) Fourth level, a processing-thread-dedicated objects collection, where an

object of said collection represents a dedicated area for data holding and data

exchange between threads (Figure 3, fourth level 94).

9.   **As to claims 19 and 20,** the claims are rejected for the same reasons as claim 5

above.

Claim Rejections - 35 USC § 103

10.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

11.   Claims 1 and 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over

Cloud et al. (hereafter Cloud)(US Pat. 6,253,369) in view of McDonald et al. (hereafter

McDonald)(US Pat. 6,053,951).

12.   Cloud was cited in the previous office action dated 12/14/2006.

13.   **As to claim 1,** Cloud discloses a computer-based method for development of

fully executable workflow applications, wherein said fully executable workflow

applications are in form of computer-executable program code for loading in computer

memory code segments and, after said computer-executable program code being

loaded and, after said workflow applications being run by invoking said program code

loaded in code segments, said program code facilitates processing of workflow orders,

wherein said method for development comprises the following steps (abstract, lines 1-

12):

(A) Describing an element of a workflow process with software that accepts

interactive user input (column 3, line 64-column 4, line 5);

(B) Interpreting accepted at step (A) user input to obtain workflow-related data

and transforming obtained data into an incremental modification of incrementally

constructed set of class objects that represent workflow process description (column 9,

lines 30-53);

(C) Repeating step (A) and step (B) in a loop until workflow-process description

is complete (Inherent in column 9, lines 30-53).

(D) Extracting workflow-process-related data from produced at step (B) set of

class objects and transforming extracted data into a workflow-process definition

(Column 16, line 56-column 17, line 44);

(E) Generating source code and compiling and linking instruction from produced

at step (D) workflow process definition (column 6, lines 31-47);

(F) Compiling generated at step (E) source code with generated at step (E) compiling instructions (column 6, lines 43-47);

(G) Linking result of performed at step (F) compiling with generated at step (E) linking instructions (column 18, lines 52-57).

14.   Cloud does not disclose that the method is a graphical method. However, McDonald discloses a method for graphical development of fully executable workflow applications (Abstract, lines 1-14 disclose a computer-implemented system and method to generate graphical code, thus can be used in the method for developing workflow applications.).

15.   Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Cloud, by utilizing a method for graphically developing code, as taught by McDonald, for the benefit of effectively providing visualization to the user, thus being able to utilize a powerful tool and efficiently being able to produce a graphical result.

16.   **As to claim 18,** the claim is rejected for the same reasons as claim 1 above.

Allowable Subject Matter

17.   Claims 2-3 and 6-17 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Furthermore, examiner has suggested amendments to the claims to overcome 101 rejections, 112 rejections, prior art, inconsistencies and informalities to put the application in condition for allowance. Attached below are examiners suggested amendments to resolve the issues at hand. Applicant is encouraged to consider examiners amendments and reply preferably by telephone, to discuss outstanding matters and amend the claims to place application in condition for allowance.

EXAMINER'S SUGGESTED AMENDMENTS

Below are the examiners suggested amendments to place application in condition for allowance. Dependent claim 2 is suggested to be cancelled after incorporating its limitations into independent claim 1. Dependent claim 6 is suggested to be cancelled after incorporating its limitations into independent claim 5. Dependent claim 18 is suggested to be amended as an independent claim, and dependent claims 19 and 20 are suggested to be cancelled as they recite limitations recited in previous claims.

1.   A computer-implemented method for graphical development of fully executable workflow applications, wherein said fully executable workflow applications are stored in a form of computer-executable program code in computer memory, a processor runs the workflow operation by invoking said program code, said program code facilitates

processing of workflow orders, wherein said method for graphical development comprises the following steps:

A. Describing an element of workflow-process graphically with software that accepts interactive user input;

B. Interpreting the user input to obtain workflow-related data and transforming obtained workflow-related data into an incremental modification of incrementally constructed set of class objects that represents a workflow-process description;

C. Repeating steps A and B in a loop until workflow-process description is complete;

D. Extracting workflow-process-related data from the set of class objects and transforming the extracted workflow-process-related data into a workflow-process definition;

E. Generating source code, compiling instruction, and linking instruction from the workflow-process definition;

F. Compiling the source code with the compiling instructions;

G. Linking results of the performed compiling with generated linking instruction; and

wherein said method further comprises defining a workflow-process comprising the following steps:

a. Establishing a matrix of workflow-activities, wherein said workflow-activities are individual items of work comprised by a potential workflow-process where all transactional operations potentially performed by each one of said individual items are potentially executed as a single transaction within each one of said workflow-activities of said potential workflow-process, and wherein said defining the matrix further comprises:

(i) Arranging workflow-activities comprised by the workflow-process being defined according to a required sequence of execution of said workflow-activities within said workflow process and according to possibilities for concurrent execution of some of said workflow-activities within said workflow-process;

(ii) Defining a constant, named PROCESS_STEPS, representing a number of sequential steps of execution of a workflow process being defined, wherein on each one of said sequential steps of execution either

one activity is executed, or either two or more activities are executed concurrently;

(iii) Defining a constant, named MAX_STEP_DEPTH representing a maximum number of workflow-activities executed concurrently on one step of workflow-process being defined;

(iv) Declaring and initializing a two-dimensional array of Boolean elements being a workflow-activities matrix, with dimensions MAX_STEP_DEPTH by PROCESS_STEPS, wherein an array element with value TRUE represents existence of a workflow-activity on position in said workflow-activities-matrix with coordinates equal to said array element indexes and wherein an array element with value FALSE represents non-existence of a workflow-activity on position in said workflow-activities with coordinates equal to said array element indexes;

b. Defining a main flow-graph, wherein the defining matrix of workflow-activities are further being represented as nodes of said flow-graph by incorporating control-connectors in order to represent a potential flow-of-control between said nodes within a potential workflow process, wherein each said control-connector signals successful execution of its sourcing workflow-activity with execution result having value TRUE, wherein two or more control-connectors sourcing from one of said nodes initiate concurrent execution of destination nodes of said control-connectors, wherein two or more control-connectors having a common destination node impose a requirement for synchronizing conjunction of signals of all control-connectors incoming to said common destination node in order to trigger execution of said common destination node, wherein said defining the main flow-graph further comprises:

(i) Defining a constant, named MAX_CONNECT_OUT, representing maximum number of main control-connectors sourcing from one flow-graph node representing a workflow-activity in a workflow-process being defined;

(ii) Declaring and initializing a two-dimensional array of integer elements with dimensions MAX-CONNECT_OUT by 2 per workflow-activity for each one of activities belonging to workflow steps with numbers from 1 to (PROCESS_STEPS-1), wherein values of each pair of elements of said array represent indexes of the initialized two-dimensional array of a workflow-activities array and thereby describe a position of a workflow-activity in said workflow-activities matrix, wherein said described positions of MAX_CONNECT_OUT workflow-activities in said workflow-activities matrix are positions of destination workflow-activities of MAX_CONNECT_OUT control connectors having common source workflow-activity with potential execution result having value TRUE;

c. Defining one or more alternative control connectors, wherein each one of said alternative control connectors signals successful execution of its sourcing workflow-activity with execution result having value FALSE, wherein two or more of said alternative control-connectors sourcing from workflow-activity initiate concurrent execution of destination workflow-activities of said alternative control-connectors, wherein a workflow-activity having established during the defining main flow-grpah a requirement for synchronizing conjunction of signals of all incoming to it control-connectors cannot be a destination workflow-activity of an alternative control connector, wherein defining the one or more alternative control-connectors further comprises:

(i) Defining a constant, named MAX_ALTCONNECT_OUT, representing a maximum number of alternative control-connectors sourcing from one flow-graph node representing a workflow-activity in a workflow process being defined;

(ii) Declaring and initializing a two-dimensional array of integer elements with dimensions MAX_ALTCONNECT_OUT by 2 per workflow activity for each one of activities belonging to workflow steps with numbers from 1 to (PROCESS_STEPS-1), wherein values of each pair of elements of said array represent indexes of the initialized two-dimensional array of a workflow-activities array and thereby describe the position of a workflow-activity in said workflow-activities matrix, wherein said described positions of MAX_ALTCONNECT_OUT workflow-activities in said workflow activities matrix are positions of destination workflow-activities of MAX_ALTCONNECT_OUT control-connectors having common source workflow-activity with potential execution result having value FALSE;

d. Defining a workflow-components matrix, wherein every element of said workflow-components matrix represents a software component, associated with a workflow-activity of a workflow-process being defined, for potential plugging for execution as part of a potential single transaction within said workflow-activity of the workflow process, wherein the defining of the workflow-components matrix further comprises:

(i) Declaring a two-dimensional array of elements of type UUID (Universal Unique Identifier, interchangeable with the term GUID, Globally Unique Identifier) with dimensions identical to dimensions of the declared two-dimensional array representing workflow-activities matrix;

(ii) Initializing elements of the declared array in the following manner: for every element of said array representing the workflow-activities matrix with value TRUE to be initialized corresponding to corresponding element with identical indexes in array representing workflow-components matrix with

UUID value representing a software component to be associated with
represented workflow-activity, and for every element of said array
representing workflow-activities matrix with value FALSE to be initialized
corresponding element with identical indexes in array representing
workflow-components matrix with UUID value indicating a dummy UUID.

2.   (Claim 2 cancelled)

3.   A computer-implemented method for graphical development of fully executable
workflow applications method as per claim 1, said method further involves interpreting
workflow-process definition data and a parameterized executing of functions that
provide software source code building blocks, wherein said parameterized executing of
functions comprises:

   a. Executing of functions that build software source code skeleton;

   b. Executing of a function that builds part of software source code that declares a
set of variables and initializes these variables with relevant data from said workflow
process definition;

   c. Executing of a function that builds part of software source code verifying that
   the necessary workflow components according to the workflow-process definition
   data are registered and available on a computer where the workflow application
   will be potentially executed;

   d. Executing of a function that builds part of the software source code
   responsible for the construction of class objects that represent workflow
   configuration according to the workflow process definition data;

   e. Executing of a function that builds compiling and linking instructions.

4.   (Claim 4 cancelled)

5.   A software product stored in a computer storage media, executed by processor
for transactional processing of workflow comprising:

   A. System skeleton as a hierarchical tree of class objects with capacity to
   represent a variety of workflow configurations, wherein:

- Said class objects are computer memory instances of classes or structures;

- Any level of said hierarchical tree contains one or multiple collections and each said collection contains a predefined or dynamically defined number of one or multiple objects, wherein objects belonging to any said collection of same hierarchy level are computer memory instances of classes or structures of identical type;

- First level of said hierarchical tree consists of a single collection and every next level of hierarchy contains one or multiple collections and individual objects, wherein said collections and said individual objects are accessible via references controlled by objects belonging to collections from previous level;

Wherein said hierarchical tree of class objects contains the following levels and siblings:

(i)　　First level, a steps-collection, wherein collection objects represent steps of said workflow-process, each said object being parent of a Second level collection;

(ii)　　Second level, an activities-collection per step of said workflow-process, wherein said collection objects represent workflow-activities that are executed concurrently with all other workflow-activities represented by objects belonging to the same collection, wherein each object is a parent of Third level siblings, said siblings being of 3 distinct categories: a processor collection, a workload-balancer object, and a workflow synchronization object;

(iii)　　Third level's sibling one, a processor collection per workflow-activity, wherein an object of said collection represents an individual workload-processing squad having a capacity to process a portion of entire workload specific for said workflow-activity, each said object being parent of a Fourth level collection;

(iv)　　Third level's sibling two, a workload-dispatching object per workflow-activity;

(v)　　Third level's optional sibling three, a workflow-synchronizing object per workflow-activity if required by flow-graph of said

workflow-process, wherein said object contains data defining
required synchronization scheme;

(vi)   Fourth level, a processing-thread-dedicated objects
collection, where an object of said collection represents a dedicated
area for data holding and data exchange between threads;

b. A means for splitting of a control flow by launching one or more new parallel
control flows, wherein said parallel control flow comprises execution of at least one
workflow-activity or sequence of plurality of workflow-activities before joining with its
launching, or other, flow of control wherein said parallel control flow launch one or more
new parallel control flows, wherein a workflow-activity that is part of a parallel control
flow generates a notification message to synchronizing a thread at control flow-join
point with other control flow or plurality of control flows about a potential result with
value FALSE of its execution, wherein a workflow-activity that is part of a parallel control
flow further splitting one or more times, generates notification messages to
synchronizing threads at splitting branches join-points with other control flows about a
potential result with value FALSE of its execution, wherein synchronizing threads
generate notification messages to synchronizing threads responsible for join-point of
synchronized control flow with one or more control flows or to synchronizing threads
responsible for join-points of said synchronizing control flows splitting branches about
termination of said synchronized control flow.

6.   (Claim 6 cancelled)

7.   A software product as per claim 5, further including means for synchronization of
two or more parallel control flows before execution of next in flow-graph workflow-
activity according to a synchronization scheme, wherein said synchronization scheme is
a conjunction of events signaling completed execution of all workflow-activities
represented by sourcing nodes of two or more flow-graph control connectors with
destination node being said workflow-activity that will be executed after said
synchronization as part of a potential single workflow-request, wherein implementation
of said synchronization method applies to entire multitude of concurrently processed
potential workflow-requests.

8.   A software product as per claim 7, further including means for launching of
alternative control flow routes, wherein said launching takes place where potential
execution of a workflow-activity has a result with value False, wherein said alternative
control flow routes redirect workflow execution from its normal routes by transferring
execution control to one or more workflow-activities with non-synchronized execution,

wherein an alternative control-connector launching an alternative control flow route and bypassing one or more workflow-activities with synchronized execution, is coupled with notification-connectors having same source node as said alternative control-connector, wherein each one of said notification-connectors has one of bypassed workflow-activities with synchronized execution as its destination node.

9.   A software product as per claim 8, further including construction of hierarchical structure of threads with four levels providing capacity for concurrent processing of multitude of workflow instances, wherein in said hierarchical structure levels below top level are organized as multitude of horizontally arranged divisions, wherein each said division is autonomous and self-contained in conducting its tasks, wherein top level thread is responsible for making adaptive decisions, and executing and supervising adaptive behavior related to allocation and de-allocation of computer and operating system resources based on its own assessment of application needs, wherein said capacity for concurrent processing of multitude of workflow instances is limited only by environmental factors such as availability of reserve of system memory and unused CPU power and ability of networking infrastructure to cope with generated traffic, wherein said hierarchical structure of threads contains following levels:

(A) First level, formed by created and activated processing threads, wherein threads forming this level of said hierarchical structure are directly responsible for transactional processing of requests for work and for transactional flow of control between workflow-activities;

(B) Second level, formed by created and activated supervising threads, wherein threads forming this level of said hierarchical structure are responsible for assignment of requests to individual processing threads and supervision of requests' execution;

(C) Third level, formed by all dispatching or synchronizing-dispatching threads being created and activated according to number of workload-dispatching objects in hierarchical tree of class objects and number of optional workflow-synchronizing objects in hierarchical tree of class objects, wherein dispatching threads being part of Third level are responsible for dispatching of requests to supervising threads, wherein synchronizing-dispatching threads being part of Third level are responsible for synchronization of execution of parallel control flows and for dispatching of requests to supervising threads;

(D) Fourth level, wherein Fourth level is hierarchy's top level and comprises only one thread being responsible for making, executing and supervising decisions about allocation and de-allocation of system resources based on its own assessment, wherein said allocation and de-allocation takes form of modification of First and Second levels of

hierarchical structure of threads and their corresponding objects and collections of hierarchical tree of class objects.

10.   A method for transactional plugging of software components into a workflow-activity of a workflow process at runtime, wherein said transactional plugging facilitates inclusion of all transactional operations by invoking a method of plugged software component into each transaction initiated by each workflow instance for execution of said workflow-activity, wherein said method for transactional plugging comprises the steps of:

> A. Creation and initialization of processing threads, wherein during its initialization each said processing thread instantiates a non-transactional component object and sends to it an amount of data being constant between processing of individual workflow requests and being necessary to perform execution of relevant portion of workflow requests that will potentially be assigned to said processing thread;
>
> B. Instantiated the non-transactional component object instantiates its own transactional component-intercepting object, and with GUID (Globally Unique Identifier), sent to it as a parameter, instantiates a workflow-activity component-intercepting object;
>
> C. Supervising thread receives potentially arriving multitudes of single workflow requests and assigns each one of said requests to an individual processing thread of its pool;
>
> D. Processing thread having an assigned request for work, invokes a method of its non-transactional component object;
>
> E. Calling a method of its transactional-component-intercepting object, wherein said call of transactional component-intercepting object method constructs transactional component objects thereby creating a new transaction and calls a method of said transactional component;
>
> F. Calling the created new transaction, said transactional component object method calls a method of the instantiated workflow-activity component-intercepting object, wherein said method call instantiates a workflow-activity software component and invokes a method of said component.

11.   A software product as per claim 10, further including means for workload
balancing structured at two levels, wherein upper level of said workload balancing
comprises multitude of associations between a dispatching thread and multitude of
supervising threads and involves dispatching thread balancing workload between its
associated supervising threads, wherein lower level of said workload balancing
comprises multiple groupings of processing threads in pools associated with a
supervising thread per pool and involves supervising threads balancing workload
between processing threads of their associated pools.

12.   A software product as per claim 11, further including means for software
bottlenecks' prevention and neutralizing, wherein said software bottlenecks' prevention
involves encapsulation of a thread pool containing fixed number of processing threads
with a supervising thread in a processing-pipe, wherein said software bottlenecks'
neutralizing comprises construction of additional processing-pipes and inclusion of
constructed additional processing-pipes in workload balancing process related to
workflow-activity where development of bottleneck has been detected.

13.   A software product as per claim 12, further including means for detection of
conditions requiring workflow application scaling up of available capacity for processing
of concurrent workflow, wherein said automatic detection in regard to a particular
workflow-activity involves checking for conjunction of events, from all processing-pipes
associated to said workflow-activity, signaling that number of idle threads in processing-
pipe's pool reached its critical minimum.

14.   A software product as per claim 13, further including means for scaling up,
wherein said application scaling up of available capacity for processing of concurrent
workflow, wherein said scaling up is triggered at a particular workflow-activity to
counteract development of a bottleneck at that particular workflow-activity and triggered
at all workflow-activities for higher workflow processing responsiveness when workload
increases, wherein said scaling up involves creation and activation of an additional
processing-pipe and inclusion of said additional processing-pipe in workload balancing
scheme.

15.   A software product as per claim 14, further including means for detection of
conditions requiring scaling down of available capacity for processing of concurrent
workflow, wherein said detection in regard to a particular workflow-activity involves

checking for conjunction of events, from all processing-pipes associated to said workflow-activity, signaling that number of busy threads in processing-pipe's pool reached its critical minimum.

16.   A software product as per claim 15, further including means for scaling down of available capacity for processing of concurrent workflow, wherein said scaling down is triggered to counteract a detected inefficiency in use of system memory and CPU time slice allocated to threads.

17.   A software product as per claim 16, further including class objects for real-time visualization of quantity, structure, and utilization of threads forming First and Second levels of hierarchical structure of threads and said hierarchical structure's adaptation-enacted modifications of its First and Second levels, wherein said visualization serves as indicator of workload volumne, indicator of points of delay caused by distributed infrastructure, and for observation and analysis of adaptive behavior of hierarchical structure of threads.

18.   A computer storage medium, containing instructions, when executed by a processor, carry out the method of:

Accepting interactive user input for a graphical description process, the instructions further performing:

A. Describing an element of workflow-process graphically with software that accepts interactive user input;

B. Interpreting the user input to obtain workflow-related data and transforming obtained workflow-related data into an incremental modification of incrementally constructed set of class objects that represents a workflow-process description;

C. Repeating steps A and B in a loop until workflow-process description is complete;

D. Extracting workflow-process-related data from the set of class objects and transforming the extracted workflow-process-related data into a workflow-process definition;

E. Generating source code, compiling instruction, and linking instruction from the workflow-process definition;

F. Compiling the source code with the compiling instructions;

G. Linking results of the performed compiling with generated linking instruction; and

Creating workflow process definition comprising:

a. Establishing a matrix of workflow-activities, wherein said workflow-activities are individual items of work comprised by a potential workflow-process where all transactional operations potentially performed by each one of said individual items are potentially executed as a single transaction within each one of said workflow-activities of said potential workflow-process, and wherein said defining the matrix further comprises:

(i) Arranging workflow-activities comprised by the workflow-process being defined according to a required sequence of execution of said workflow-activities within said workflow process and according to possibilities for concurrent execution of some of said workflow-activities within said workflow-process;

(ii) Defining a constant, named PROCESS_STEPS, representing a number of sequential steps of execution of a workflow process being defined, wherein on each one of said sequential steps of execution either one activity is executed, or either two or more activities are executed concurrently;

(iii) Defining a constant, named MAX_STEP_DEPTH representing a maximum number of workflow-activities executed concurrently on one step of workflow-process being defined;

(iv) Declaring and initializing a two-dimensional array of Boolean elements being a workflow-activities matrix, with dimensions MAX_STEP_DEPTH by PROCESS_STEPS, wherein an array element with value TRUE represents existence of a workflow-activity on position in said workflow-activities-matrix with coordinates equal to said array element indexes and wherein an array element with value FALSE represents non-existence of a workflow-activity on position in said workflow-activities with coordinates equal to said array element indexes;

b. Defining a main flow-graph, wherein the defining matrix of workflow-activities are further being represented as nodes of said flow-graph by incorporating control-connectors in order to represent a potential flow-of-control between said nodes within a potential workflow process, wherein each said control-connector signals successful execution of its sourcing workflow-activity with execution result having value TRUE, wherein two or more control-connectors sourcing from one of said nodes initiate concurrent execution of destination nodes of said control-connectors, wherein two or more control-connectors having a common destination node impose a requirement for synchronizing conjunction of signals of all control-connectors incoming to said common destination node in order to trigger execution of said common destination node, wherein said defining the main flow-graph further comprises:

(i) Defining a constant, named MAX_CONNECT_OUT, representing maximum number of main control-connectors sourcing from one flow-graph node representing a workflow-activity in a workflow-process being defined;

(ii) Declaring and initializing a two-dimensional array of integer elements with dimensions MAX-CONNECT_OUT by 2 per workflow-activity for each one of activities belonging to workflow steps with numbers from 1 to (PROCESS_STEPS-1), wherein values of each pair of elements of said array represent indexes of the initialized two-dimensional array of a workflow-activities array and thereby describe a position of a workflow-activity in said workflow-activities matrix, wherein said described positions of MAX_CONNECT_OUT workflow-activities in said workflow-activities matrix are positions of destination workflow-activities of MAX_CONNECT_OUT control connectors having common source workflow-activity with potential execution result having value TRUE;

c. Defining one or more alternative control connectors, wherein each one of said alternative control connectors signals successful execution of its sourcing workflow-activity with execution result having value FALSE, wherein two or more of said alternative control-connectors sourcing from workflow-activity initiate concurrent execution of destination workflow-. activities of said alternative control-connectors, wherein a workflow-activity having established during the defining main flow-grpah a requirement for synchronizing conjunction of signals of all incoming to it control-connectors cannot be a destination workflow-activity of an alternative control connector, wherein defining the one or more alternative control-connectors further comprises:

(i) Defining a constant, named MAX_ALTCONNECT_OUT, representing a maximum number of alternative control-connectors sourcing from one flow-graph node representing a workflow-activity in a workflow process being defined;

(ii) Declaring and initializing a two-dimensional array of integer elements with dimensions MAX_ALTCONNECT_OUT by 2 per workflow activity for each one of activities belonging to workflow steps with numbers from 1 to (PROCESS_STEPS-1), wherein values of each pair of elements of said array represent indexes of the initialized two-dimensional array of a workflow-activities array and thereby describe the position of a workflow-activity in said workflow-activities matrix, wherein said described positions of MAX_ALTCONNECT_OUT workflow-activities in said workflow activities matrix are positions of destination workflow-activities of MAX_ALTCONNECT_OUT control-connectors having common source workflow-activity with potential execution result having value FALSE;

d. Defining a workflow-components matrix, wherein every element of said workflow-components matrix represents a software component, associated with a workflow-activity of a workflow-process being defined, for potential plugging for execution as part of a potential single transaction within said workflow-activity of the workflow process, wherein the defining of the workflow-components matrix further comprises:

(i) Declaring a two-dimensional array of elements of type UUID (Universal Unique Identifier, interchangeable with the term GUID, Globally Unique Identifier) with dimensions identical to dimensions of the declared two-dimensional array representing workflow-activities matrix;

(ii) Initializing elements of the declared array in the following manner: for every element of said array representing the workflow-activities matrix with value TRUE to be initialized corresponding to corresponding element with identical indexes in array representing workflow-components matrix with UUID value representing a software component to be associated with represented workflow-activity, and for every element of said array representing workflow-activities matrix with value FALSE to be initialized corresponding element with identical indexes in array representing workflow-components matrix with UUID value indicating a dummy UUID.

and

Generating source code and compiling and linking instructions sufficient to build
fully executable workflow applications by:

    a. Executing of functions that build software source code skeleton;

    b. Executing of a function that builds part of software source code that
    declares a set of variables and initializes these variables with relevant
    data from said workflow process definition;

    c. Executing of a function that builds part of software source code verifying
    that the necessary workflow components according to the workflow-
    process definition data are registered and available on a computer where
    the workflow application will be potentially executed;

    d. Executing of a function that builds part of the software source code
    responsible for the construction of class objects that represent workflow
    configuration according to the workflow process definition data;

    e. Executing of a function that builds compiling and linking instructions.

19.   (Claim 19 cancelled)

20.   (Claim 20 cancelled)

Response to Arguments

1.   Applicant's arguments with respect to claims 1-3 and 5-20 have been considered

but are moot in view of the new ground(s) of rejection.

Conclusion

2.   **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time

policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE

MONTHS from the mailing date of this action. In the event a first reply is filed within

TWO MONTHS of the mailing date of this final action and the advisory action is not

mailed until after the end of the THREE-MONTH shortened statutory period, then the

shortened statutory period will expire on the date the advisory action is mailed, and any

extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later

than SIX MONTHS from the mailing date of this final action.

Again, applicant is requested to contact examiner in order discuss amendments

necessary to place application in condition for allowance. Examiner contact information

is listed below.

Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Michael Yaary whose telephone number is (571) 270-

1249. The examiner can normally be reached on Monday-Friday, 8:00 a.m - 5:00 p.m..

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Meng-Ai An can be reached on (571) 272-3756. The fax phone number for

the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the

Patent Application Information Retrieval (PAIR) system. Status information for

published applications may be obtained from either Private PAIR or Public PAIR.

Status information for unpublished applications is available through Private PAIR only.

For more information about the PAIR system, see http://pair-direct.uspto.gov. Should

you have questions on access to the Private PAIR system, contact the Electronic

Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a

USPTO Customer Service Representative or access to the automated information

system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Michael Yaary
571-270-1249
Michael.Yaary@uspto.gov

MY

MENG-AI T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100